AAS-XX-XXX

EVALUATION OF PRECISION LANDING PERFORMANCE USING A GENERALIZED AEROSPACE SIMULATION IN SIMULINK® FRAMEWORK

Juan Orphee,[*] Nicholas Olson,[†] Keith Clements,[†] Matt Hawkins,[*] Alexander Summers,[†] Wei Wu,[†] Michael Fritzinger,[*] Kyle Miller[*]

NASA's science and exploration goals to return to the Moon and beyond will need to perform precision landings to place humans and cargo supplies near places of scientific interest, surface resources, or pre-established basecamps. With the maturation of new navigation technology, such as terrain relative navigation, precision landing is now feasible, enabling new exploration sites, such as the lunar poles. However, verification of precision landing performance becomes crucial since not reaching the designated landing site would have a high risk of loss of mission. Therefore, having a high-fidelity simulation platform to evaluate six degrees of freedom vehicle performance during high-risk phases of flight such landing is a fundamental part of the system verification and risk reduction. The NASA Marshall Space Flight Center has developed the GeneraLized Aerospace Simulation in Simulink® (GLASS) tool which incorporates guidance, navigation, and control algorithms, as well as vehicle and environmental models, such as gravity, vehicle mass properties, navigation sensors, propulsion, and terrain models. GLASS uses the MathWorks® Simulink® environment which provides a model-based design framework that allows the incorporation of vehicle models in a modular architecture. The Simulink® environment provides seamless integration with all the MathWorks® capabilities and toolboxes, such as control design toolboxes and Simscape™ Multibody™ dynamics toolbox. The MathWorks® environment also allows for guidance, navigation, and control algorithms to be auto coded in C language, enabling quick software and hardware in the loop testing. This paper provides an overview of GLASS capabilities for analyzing precision landing performance, including navigation trades applied to a NASA human lander reference design architecture.

[*] Aerospace Engineer, EV42, NASA/Marshall Space Flight Center, Huntsville AL 35812

[†] Aerospace Engineer, EV41, NASA/Marshall Space Flight Center, Huntsville AL 35812

INTRODUCTION

NASA's Artemis program is composed of many sub-programs such as the Space Launch System (SLS) rocket, the Orion spacecraft, the Human Landing System (HLS), Exploration Ground Systems (EGS), the lunar Gateway, and Artemis Base Camp (ABC). This paper focuses on the Human Landing System. For the HLS, NASA develops the high-level mission requirements and leaves the design solution to the industry partners. The wide-open design space leads to widely different vehicle and mission architecture solutions. Part of NASA's role is to understand and evaluate the risks of the proposed architecture and ultimately perform the Certification of Flight Readiness (CoFR). To assess multiple vehicle architectures, NASA MSFC developed the GeneraLized Aerospace Simulation in Simulink® (GLASS). This tool was originally developed during the NASA lunar Cargo Transportation and Landing by Soft Touchdown (CATALYST) initiative.

As part of the CATALYST initiative, NASA competitively selects industry partners to develop commercial cargo landers to deliver payloads to the lunar surface. GLASS was developed to perform rapid development and evaluation of the GNC performance of partner's robotic lunar lander architectures as well as NASA's own lander concept, the Lunar Pallet Lander.[1],[2],[3] After CATALYST ended, NASA shifted focus to the human scale version, with the Human Landing System program in 2019. GLASS is a multi-body Six Degrees of Freedom (6-DoF) tool that is used to evaluate the Guidance Navigation and Control (GNC) performance of one or multi-element architectures.

The GLASS tool is used for analyzing the precision landing performance for the Human Landing System NASA reference design.[4] The NASA reference design was used as a baseline for the Artemis program Appendix H Broad Agency Announcement (BAA). In May 2020, three companies competed for the development of an integrated lunar lander vehicle. And in April 2021 NASA announced SpaceX as the wining Appendix H partner. However, this paper focuses on the NASA reference design architecture, also referred as the "government reference design." The GLASS tool is used to evaluate the performance of the Guidance, Navigation and Control system for the government reference design architecture. In addition, a navigation trade study is performed as an example of some of the GLASS capabilities. In this case, the passive optical Terrain Relative Navigation system is evaluated under different operation altitudes and downrange from the landing site to mimic surface illumination constraints associated with the polar lighting conditions.

THE GLASS FRAMEWORK AND CAPABILITIES

GLASS is designed to be a modern, modular, user-friendly, and efficient 6-DoF simulation tool for aerospace vehicles. As the name denotes, GLASS is built within the MathWorks® MATLAB® and Simulink® environment. This allows engineers to utilize the pre-built and tested toolboxes to speed up their development effort. To further simplify the development work for engineers, the Simscape™ Multibody™ toolbox forms the basis for the 6-DoF equations of motion. Simscape™ Multibody™ generates the equations of motion internally and provides a multibody simulation environment using joints, bodies, force elements, and sensors. GLASS provides a layer of abstraction over the base Simscape™ Multibody™ blocks. Simplifying the building of a vehicle by combining multiple Simscape™ Multibody™ blocks into single masked GLASS blocks. These parameterized blocks ensure models are constructed uniformly regardless of the vehicle design and have been complied in a Simulink® library. This Simulink® library has taken the name of "GLASS_Core" to note that these blocks form the core of any simulation built using the GLASS framework.

GLASS Architecture

Simulations built within the GLASS framework share a common architecture. The top level of the simulation is broken into five components: Plant, Sensors, Navigation, Guidance, and Control. Each of these components resides in a separate referenced model to allow groups to build, update, and test their models and code independently of the main simulation. This architecture aids in both simulation organization as well as unit test comparisons between other simulation tools by breaking down a large simulation into individual, self-contained models. Model references enforce pre-defined inputs and outputs, which help to ensure that the different models interface correctly. Figure 1 below demonstrates the top-level organization and typical data flow.

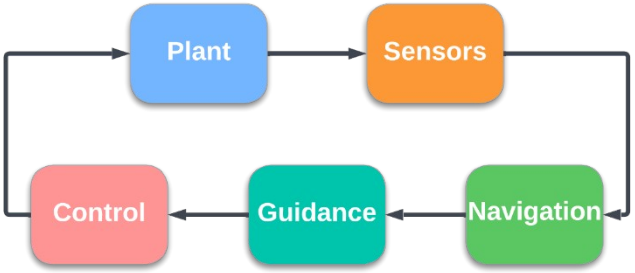


Figure 1. Top-Level GLASS Architecture.

The top-level models are further broken down internally using model references, subsystem references, and libraries to maintain a modular approach. Simulink® files are treated as binary files by source control tools such as GIT and SVN, which means that those tools are unable to diff (compare file differences) and merge those files natively. Therefore, MathWorks® provides built-in tools for doing file comparisons and merge operations. In addition, the GLASS framework breaks the large top-level models into smaller components to allow developers to work independently of one another, reducing interference, while maintaining simulation functionality. The result is a faster and smoother simulation update cycle. The layout of the Plant model will be discussed next.

The Plant model contains the bulk of the GLASS components for a given simulation, including all the components that are based on Simscape™ Multibody™. The Plant is further broken into two separate model references: plant models and 6-DoF dynamics. All the blocks relating to the Simscape™ Multibody™ toolbox are contained within the 6-DoF dynamics model. This model is then converted to a protected plant model, which eliminates the need for any license other than a standard Simulink® license for developers running the simulation. The 6-DoF dynamics model receives mass properties, forces, and torques as inputs while outputting vehicle states. This architecture places models such as thrusters, engines, mass properties, disturbances, etc. outside of the protected plant. In doing so, the protected 6-DoF model is seldom updated once the initial Simscape™ Multibody™ model has been created. This minimizes the number of licenses required. Developers update models contained in the plant models portion of the plant and the outputs directly feed the 6-DoF dynamics model of the vehicle. Figure 2 displays the layout of the plant model.

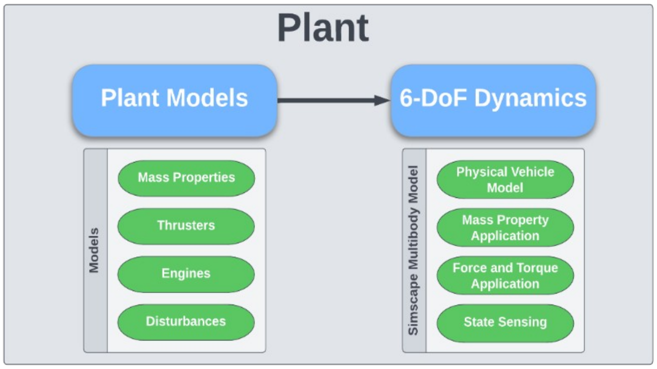


Figure 2. GLASS Plant Framework.

GLASS also contains a library of functions for operations such as quaternion math, orientation conversions, and unit conversions. This reduces the need to rely on possibly limited availability toolboxes such as the Aerospace Blockset™. Along with these functions and the previously discussed core dynamics blocks, the library also contains sensor models, gravitational models, and slosh models. These models are parameterized to allow developers to quickly pull them into a new GLASS simulation without having to start from scratch developing a new model. Further effort is being put into developing parameterized thruster and disturbance libraries among others. GLASS also utilizes JPL's SPICE ephemeris routines for computing planetary ephemeris positions and orientations.[5]

Multibody Framework

Simscape™ Multibody™ forms the core foundation of GLASS simulations. Compared to other simulation tools that directly code the equations of motion (EOM) for the vehicle, GLASS utilizes the Simscape™ Multibody™ toolbox to handle the EOM automatically. Since Simscape™ Multibody™ is based in Simulink®, it uses a graphical approach to construct a 6-DoF simulation using joints, transforms, force and torque applications, and mass property applications. Developers utilize these basic components to assemble a vehicle.[6]

Rather than working directly with the basic Simscape™ Multibody™ blocks such as joints and transforms, GLASS provides a layer of abstraction from these blocks to allow developers to quickly assemble a simulation. The GLASS library blocks handle manipulating the base Simscape™ Multibody™ blocks and present the user with clearly defined masks and inputs. A basic aerospace vehicle simulation can be constructed with as few as six GLASS library blocks as shown in Figure 3.

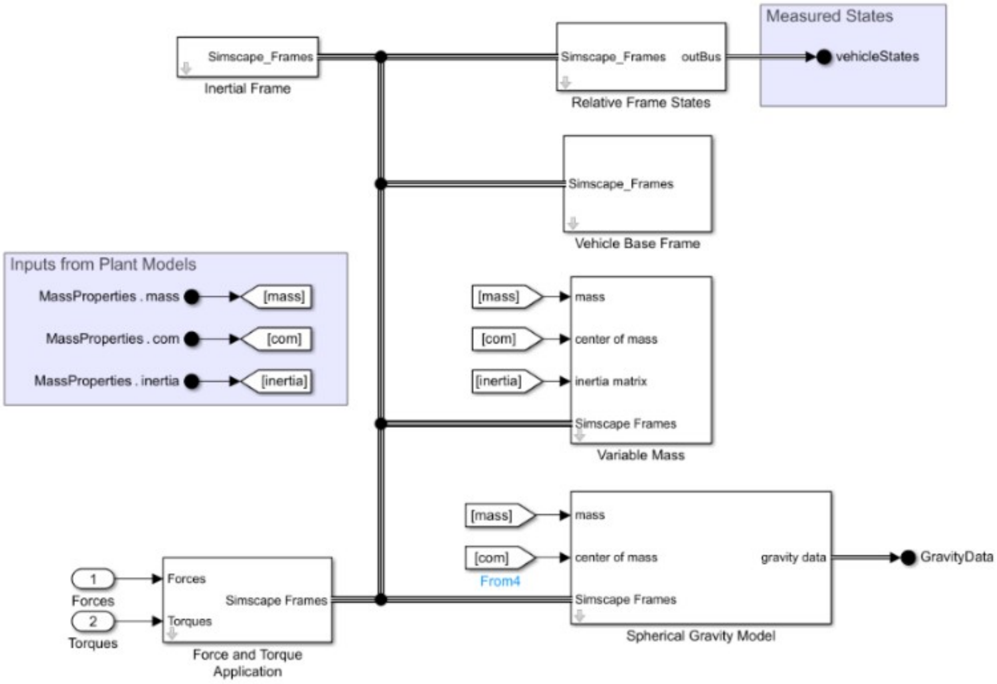


Figure 3. Simple GLASS 6-DoF Dynamics Model.

Auto-coding and Hardware in the Loop Capabilities

GLASS supports both the auto-coding of GNC algorithms as well as testing these algorithms in a Hardware In the Loop (HIL) scenario. GLASS was utilized on both the Lunar Pallet Lander, as well as the Lunar CATALYST program on the Masten XL-1T lander. Both programs utilized Core Flight Software (cFS) as the flight software for the lander. cFS provides a Simulink® Interface Layer (SIL) code package that allows for the code generator to produce flight ready code. NASA's MSFC produced a set of GNC code for the XL-1T lander to be used for testing the flight processor to command set of propulsion valves. NASA's JSC provided the cFS code for the processor along with integration assistance with the auto-coded GNC algorithms. In this test setup, a single board computer executing flight software interacted with a Speedgoat (a Performance Real-Time Target machine) executing a model of the vehicle compiled from the GLASS simulation. This vehicle model is executed in real time and emulates the sensor suite intended to be installed on the vehicle along with the full vehicle dynamics. The test showed the successful auto generation of code, integration, actuation of actual flight hardware, and the real time operation of a GLASS simulation on the Speedgoat hardware.

Contact Dynamics and Landing Stability Analysis Capability

Stable vehicle dynamics during touchdown is key for a successful landing. The landing leg subsystem plays a critical role in the touchdown vehicle dynamics, affecting landing success. During a landing event, several factors play an important role that determine the vehicle landing stability during touchdown. Some of those factors are the vehicle's touchdown velocities, the vehicle's center of gravity height relative to landing leg footprint area, the terrain slope and roughness, and the landing leg energy absorption capabilities.

The landing gear is modeled in GLASS and is comprised of the primary strut, secondary struts, footpad, and joints, that are easily modelled using the Simscape™ Multibody™ toolbox. The energy absorption mechanisms in the struts are readily modelled using Simulink® and Multibody blocks based on the relative motion of the inner and outer cylinder of the struts. Simscape™'s landing leg model is given in Figure 4-a. During the touchdown, the footpads strike the lunar surface. The contact forces can be modeled using the Spatial Contact Force block, where the soil mechanical properties are specified. Slosh dynamics are also included using a mechanical pendulum model. Figure 4-b shows an example of a stable landing analysis on a 4.6 degree slope for the 2-2-over landing mode, where two landing legs touch first and the second set of landing legs touch second over terrain that is sloping away.[7] Figure 4-c, shows the stroke length vs. time for all legs, legs 1 and 4 (rear legs) touching down first and legs 2 and 3 (front legs) touching second. The stroke time history can be used to ensure the attenuation mechanism is not over stroking during landing.

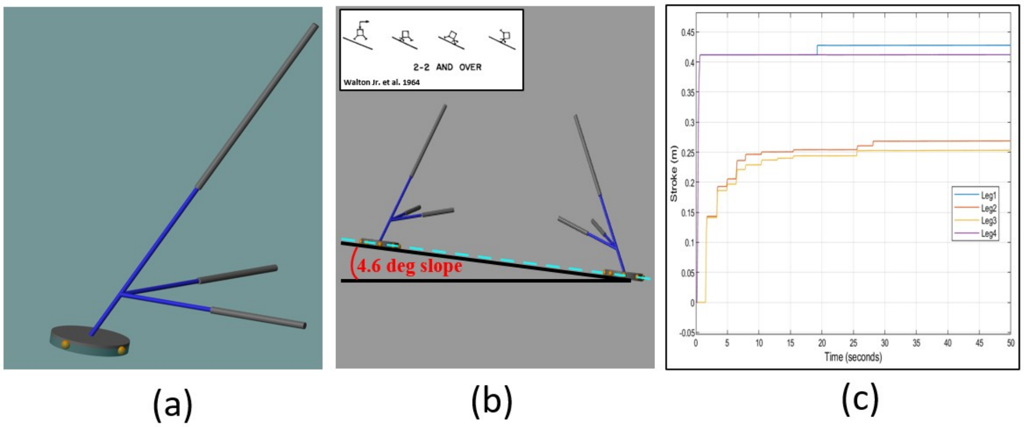


Figure 4. (a) Leg Structure, (b) Landing on Sloped Terrain, (c) Landing Stroke Length.

HUMAN LANDING SYSTEM REFERENCE DESIGN

NASA developed a government reference design for the Human Landing System to allow NASA to perform early trades and feasibility studies to assess and communicate risk of a particular design choice or mission constraint. The present analysis will focus on the government reference design, NASA Design Analysis Cycle (DAC).[4] The design at landing is a two-element architecture with a Descent Element (DE) and Ascent Element (AE) as shown in Figure 5. Table 1 lists relevant vehicle information and environment modeling assumptions, including ephemeris and gravity models. [5],[8]

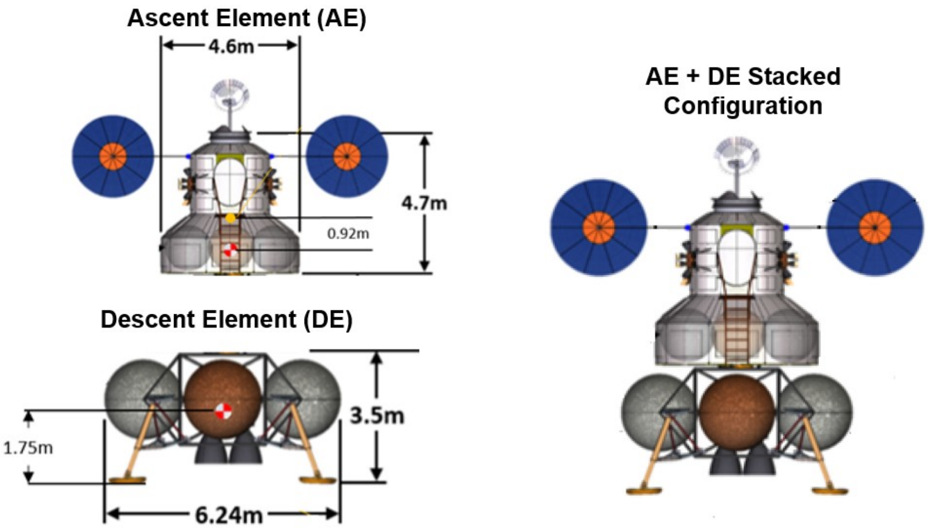


Figure 5. Government Reference Design.

Table 1. Vehicle Information.

Mass	Descent Element Total Mass	30,392 kg
	Descent Element Propellant Mass	26,324 kg
	Ascent Element Total Mass	12,172 kg
Propulsion	Descent Element Main Engines	3x 8000 lbf @ 340 s ISP
	Descent Reaction Control System	16x 100 lbf @ 300 s ISP
Environment	Lunar Gravitational Model	Zonal Harmonic J2
	Ephemeris Kernels	Sun, Earth, Moon

Deorbit Descent Landing for a Human Lander Government Reference Design

The Deorbit Descent Landing (DDL) sequence of events for the Artemis Lunar lander Government reference design is shown in Figure 6. The De-Orbit Insertion (DOI) maneuver is initiated at the 100km parking orbit, and the burn takes the vehicle to a 100x15.24 km orbit. Following the DOI there is a coast period to the Powered Descent Initiation (PDI) braking burn, performed by the DE at the periapsis. The braking burn is designed to reduce the vehicle velocity as efficiently as possible. The vehicle then transitions to the approach phase, during which the vehicle orientation is adjusted such that landing navigation sensors and crew visibility to landing site are accommodated. During this phase, the vehicle thrust is controlled such that once the vehicle is over the landing site, horizontal velocity is nullified. Note that the “braking” and “approach” nomenclature is adopted from the Apollo Moon landings. Terminal vertical descent phase begins at an altitude of 50 m. The vertical descent starts with a 5 m/s downward velocity that is linearly reduced with altitude such that the main engines cut off an altitude of 1 m and a velocity of 1 m/s.

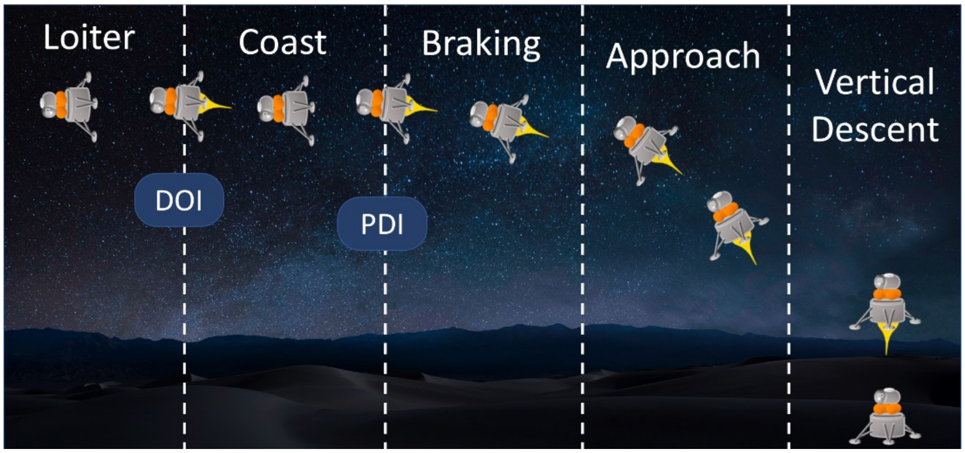


Figure 6. Deorbit Descent Landing Phases.

The GLASS simulation is phase-based, with different guidance algorithms in each phase, and different criteria for changing phases. The descent flight phases, and corresponding descriptions are shown in Table 2. Details of phase transitions and guidance algorithms for each phase follow later.

Table 2. List of DDL Phases.

Phase Name	Description
Pre-Burn Coast/Loiter	Vehicle commands desired attitude for de-orbit burn with zero throttle
De-Orbit Insertion Burn	Vehicle commands thrust in the anti-velocity direction
Pre-Braking Coast	Vehicle commands desired attitude for braking burn with zero throttle
Braking	Vehicle uses Apollo-derived guidance to remove most of the vehicle’s

	velocity
Approach	Vehicle uses Apollo-derived guidance to reach a target directly above the landing site
Vertical Descent	Vehicle nulls position and velocity errors while touching down

GUIDANCE

This paper documents the phases used in the government reference design, and the triggers to transition between phases. A variety of triggers are used to transition between flight phases, including measured trajectory variables and burn time computed by the guidance algorithms. Table 3 lists the descent flight phases, the guidance algorithms for each phase, and a description of the conditions that trigger each new flight phase.

Three main guidance algorithms are used for DDL. The initial De-Orbit Insertion (DOI) burn is accomplished by simply commanding thrust along the anti-velocity direction until the measured perigee reaches a target value. The braking and approach burns use the Fractional-Polynomial Powered Descent Guidance (FP^{2}DG) algorithm develop by Lu.[9] This algorithm is similar to the Apollo guidance algorithms, with variable gains for trajectory shaping. In practice, it has been found that selecting gains to make the braking burn equivalent to Apollo E-Guidance and the approach burn equivalent to Apollo landing guidance works well for dispersed Monte Carlo simulations. Finally, vertical descent uses a proportional-derivative (PD) scheme to null horizontal position and horizontal velocity errors, while using a linear ramp down of the vertical velocity to the final value for landing.

The FP^{2}DG law uses the current state (position **r** and velocity **v**), the desired final state (**r**$_f$, **v**$_f$, and acceleration **a**$_f$), and the acceleration due to gravity (**g**). Equation (1) shows the Apollo E-Guidance formulation, used for the braking burn. Equation (2) shows the Apollo landing guidance formulation, used for the approach.

$$a_{brake} = \frac{-2}{t_{go}}\left[v_f - v\right] + \frac{6}{t_{go}^2}\left[r_f - r - v\,t_{go}\right] - g \qquad \textbf{(1)}$$

$$a_{approach} = a_f - \frac{6}{t_{go}}\left[v_f - v\right] + \frac{12}{t_{go}^2}\left[r_f - r - v\,t_{go}\right] \qquad \textbf{(2)}$$

The error-nulling scheme for vertical descent applies thrust mainly in the vertical (**z**) direction, with gains, K_P and K_D, on the position and velocity errors, respectively. Note that in landing site coordinates, the current position and velocity are equivalent to the position and velocity errors. Finally, the magnitude of the acceleration command is scaled to linearly ramp down over time. Equation (3) shows the error-nulling scheme.

$$a_{vert} = \frac{k_P r + k_D v + \hat{z}}{v(ramp)} \quad \textbf{(3)}$$

Table 3. List of Phase Transitions.

Phase Name	Description	Starting Trigger	Trigger Details
Pre-Burn Coast/Loiter	Thrusters in anti-velocity direction (zero throttle)	N/A	Initial condition
De-Orbit Insertion Burn	Burn in anti-velocity direction	$t \leq t_{burnstart}$	Fixed start time
Pre-Braking Coast	Compute attitude for powered descent, zero throttle	$r_{peri} \leq r_{peri,target}$	Perigee below target value
Braking	FP^{2}DG tuned to match Apollo E-Guidance	$r_{LS} \leq r_{LS,target}$	Range to Launch Site below target value
Approach	FP^{2}DG tuned to match Apollo landing guidance	$t_{go} \leq 0$	Computed time-to-go reaches zero
Vertical Descent	Vehicle nulls position and velocity errors with velocity ramp down	$t_{go} \leq 0$	Computed time-to-go reaches zero

CONTROLS

Many space-based projects and missions make use of a common set of mechanical and propulsive actuators. GLASS takes advantage of this repetition by maintaining a toolbox of generalized controllers that can be applied to these actuator systems. Some common controllers provided in the GLASS libraries include a Proportional Integral Derivative (PID) controller for a Thrust Vector Control (TVC) system and phase-plane type controller for a Reaction Control System (RCS).

Single-Engine TVC Controller

Thrust Vector Control (TVC) is a common propulsion-based control system. In this system, linear actuators deflect the thrust force line of action to either side of the vehicle center of mass to create vehicles torques. A TVC controller needs to command the correct deflections of the nozzle to achieve desired vehicle states with acceptable stability. GLASS supplies a general and flexible PID controller tool for single-engine TVC control. The controller commands rock and tilt deflections to create torque responses of a single nozzle when provided with pointing and angular rate error commands. The PID gains can be updated to the desired tuning, and the tool offers many common PID supplements such as integrator wind-up protection and output saturations.

Multiple-Engine TVC Controller

It is not uncommon for launch vehicles to use multiple TVC systems simultaneously. GLASS provides an additional TVC tool for the commanding of over actuated TVC systems. The tool applies a PID and a pseudo-inverse allocation method to solve for multiple nozzle deflections as detailed by Orr et al.[10] The PID controller commands the desired vehicle angular acceleration based on pointing and angular rate error inputs. Those desired angular acceleration are passed to the allocator which uses estimates of vehicle mass properties and engine locations to solve for the needed multi-engine TVC deflections to achieve the desired angular accelerations. GLASS implements the pseudo-inverse allocator method shown in Figure 7. This algorithm provides a general approach to achieve multiple-engine TVC control and can be updated easily with new vehicle properties.

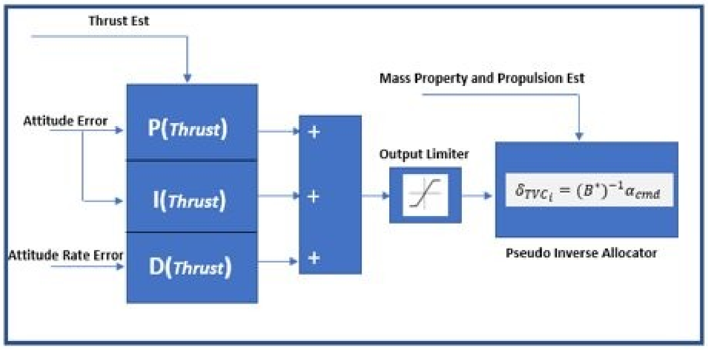
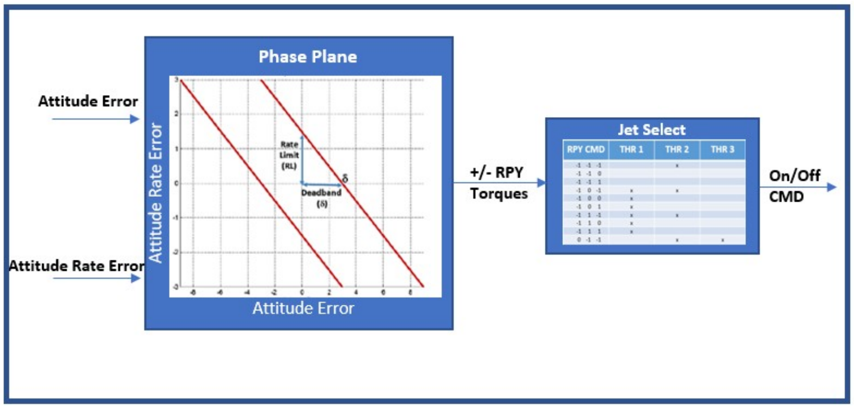


Figure 7**.** Simple PID Diagram and Control Allocator.

RCS Thruster Phase Plane Controller

Reaction control thrusters are a non-linear system and require non-linear control methods. GLASS offers a generalized phase-plane controller where the commanding of an RCS thruster is dictated in reference to pointing error and angular rate error. These errors are compared against an on/off threshold logic that is designed to maintain the vehicle dynamics within acceptable errors implemented following Hall et al. method in Figure 8.[11] The tool allows for easy alteration of its input parameters enabling a flexible design for new spacecraft and operations using RCS thrusters.



RPY CMD	THR 1	THR 2	THR 3
-1 -1 -1		x	
-1 -1 0			
-1 -1 1			
-1 0 -1	x	x	
-1 0 0	x		
-1 0 1	x		
-1 1 -1	x	x	
-1 1 0	x		
-1 1 1	x		
0 -1 -1		x	x

Figure 8. Simplified Phase Plane Controller.

The GLASS team encounters new and more intricate control systems through the natural flow of projects and vehicle design cycles. Engineers in the GLASS workflow take the opportunity to generalize these controllers and build up GLASS’s control toolbox libraries. This flow streamlines the control and trade study work of its members and allows for efficient turnover of rapid designs.

NAVIGATION

GLASS contains a sophisticated navigation and sensor suite suitable for a wide range of scenarios. Users can easily configure hardware and navigation filter parameters depending on the mission. Sensor coverage for lunar descent is shown below in Table 4. The descent simulation begins with Deep Space Network (DSN) observations prior to the Deorbit burn phase. The ground station DSN navigation update is treated as a direct navigation state update and covariance reset at simulation time zero. The star tracker supplies absolute attitude observations during On-Orbit and Coast phases but remains inactive during burns due to the dynamic nature of the burning phases. The Terrain Relative Navigation (TRN) activates late in the Coast phase, supplying absolute position observations from directly prior to the Braking phase until minimum altitude or pointing constraints are violated late in the braking phase, deactivating the TRN sensor. Thereafter, the Navigation Doppler Lidar (NDL), used for altimeter and velocimeter measurements, is utilized roughly midway through the approach phase until the start of vertical descent. The IMU is supplying observations for the entire flight, but accelerometer measurements below a noise threshold are rejected while on orbit or in other quiescent phases of flight. Rejecting low IMU accelerometer measurements ensures that the navigation propagator doesn’t incorporate noisy acceleration data during quiescent periods in which the vehicle accelerations are known to be below the noise of the IMU. Measurement rejection for the gyros based on a minimum threshold is not utilized because star tracker corrections are available during each quiescent coast phase.

Table 4. Navigation Sensor On-Time by Phase

Coast to DOI	DOI	Coast to PDI	Braking Phase	Approach Phase	Terminal Descent
IMU + Star Tracker + DSN	IMU	IMU + Star Tracker + TRN	IMU + TRN	IMU + NDL	

Navigation Filter and Sensors

The navigation filter consists of two sequentially executed Extended Kalman Filters (EKF), one for attitude state estimation and the other for translational state estimation. For attitude, a 6-state EKF filter is used which estimates attitude and gyro bias. For translation, a 12-state filter estimates position, velocity, accelerometer bias error, and accelerometer scale factor error. The filters fuse measurements from the IMU, star tracker, TRN and NDL as well as DSN as they become available, depending on flight phase. Star tracker observations provide highly accurate attitude updates, while TRN and NDL provide position and velocity corrections. The filter utilizes a J2 zonal harmonic model of the moon and uses first order approximations for the Earth and sun's gravity. Tuning of the filter is based on sensor performance models which are discussed in the following section. Other sensor models within GLASS, but outside the scope of this paper, include radar sensors, ground and orbiting Radiometric Frequency (RF) beacon assets, Laser and optical range sensors, and Global Positioning System (GPS) measurements.

IMU Model

Pulling from Marshall's heritage with launch vehicle and spacecraft sensor modeling and simulation, GLASS utilizes a generalized parametric IMU model capable of capturing up to 36 key IMU performance errors for spacecraft and launch vehicles. The model uses high-rate truth dynamics at the vehicle center of gravity mapped to the sensor mounting location as an input. A comprehensive noise model along with other error models for biases, misalignments, scale factors and latencies among other things corrupt these truth observations which are used in propagation of the navigation state. Past test efforts at MSFC have shown satisfactory agreement between this parametric sensor model and actual IMU performance during 72-hour static and short duration dynamic tumble tests using various performance class IMUs ranging from Micro-Electro-Mechanical Systems (MEMS) to navigation grade quality.

Star Tracker Model

GLASS utilizes a parametric star tracker model which relies on truth observations which are corrupted based on a generalized error model approach. For the star tracker error model, absolute inertial attitude observations are received at a set frequency defined by the user. From there, the Field of View (FOV) is assessed for interference with keep out zones on the vehicle and in the night sky. If the measurement is deemed possible by these checks, error sources for cross and along boresight noise, as well as misalignments and biases due to sensor performance and environmental effects are incorporated. Capturing of the Star Tracker locations and orientations on the vehicle, especially for lunar descent missions, is critical to determining coverage over key phases of flight and is another user defined parameter for this model.

Navigation Doppler LiDAR

The Navigation Doppler LiDAR (NDL) is a sensor that provides line-of-sight range and doppler velocity measurements relative to the planetary surface. Using three beams, the NDL sensor allows navigation filter to accurately estimate altitude and velocity vector with respect to the planet surface. Estimating altitude and velocity also indirectly reduces the rate of horizontal position error accumulation. The true range to the local terrain along the boresight of each NDL beam is determined using an efficient ray tracing algorithm presented in Shidner et al. [12] True line-of-sight velocity is determined by projecting the planet-relative velocity of the sensor onto the direction of each beam. These truth range and velocity values are then passed to an error model created by the NDL development team at NASA’s Langley Research Center (LaRC).[13] The model considers laser characteristics (frequency linewidth, frequency knowledge, frequency drift), beam pointing knowledge, and dynamic effects (acceleration, angular rates, vibrations) when determining sensor errors. The sensor errors are added to the truth values and the measurement is sent to the translational EKF.

Terrain-Relative Navigation

A parametric Terrain-Relative Navigation (TRN) sensor model was implemented to represent a passive optical (camera-based) TRN system onboard the lander. TRN can directly measure position and thus eliminate a large amount of the navigation positional error growth that accumulates since the last DSN state update prior to the DOI burn. Without TRN, a precise landing (<100m from a target) is unlikely. The TRN model implemented for the government design simulation is a simplified attempt to capture the behavior of a real TRN system without the need for rendering imagery, a process that can be time consuming. An actual TRN system would (1) take a camera image of the planet/moon surface, (2) pull out features from that image, (3) compare those features to on-board feature maps, and (4) use those matches in aggregate to estimate the vehicle state. By finding matches between the collection of features in the image taken and the features in the on-board map, the TRN computer processor can estimate the vehicle’s position and orientation, with the position measurement being the most useful navigation state update. However, the model implemented for this project directly generates feature matches randomly within the camera field-of-view and uses those randomly generated matches to determine the position measurement. The model adds several error sources to simulate the image processing errors of a real system, including feature location errors in the camera image and the on-board map, cross-correlation error when matching the feature in the image to the on-board map, map quantization error, regional feature location bias errors, outlier errors, global map tie errors, and vehicle altitude uncertainty errors. The estimation algorithm used in the TRN model, which converts feature matches into a state estimate, is an iterative Extended Kalman Filter (EKF) based on the formulation in Mourikis et al.[14] This formulation was also used for the fine matching mode of the Lander Vision System (LVS) on the Mars 2020 lander.[15]

Nominal Sensor Parameters. The baseline sensor parameters used by the models presented in this paper are listed in Table 5. These values represent the nominal operation of each model.

Table 5. Nominal Sensor Parameters.

Sensor	Parameter	Value
IMU	Measurement Frequency	100 Hz
	Measurement Threshold Variance	4.2e-5 m/s
NDL	Operational Range	30 - 3000 m altitude
	Measurement Frequency	20 Hz
Star Tracker	Measurement Frequency	10 Hz
TRN	Quality	High
	Measurement Frequency	1 Hz
	Operational Range	16.0-1.5 km altitude
	Field of View	90°
	Observable Landmarks	100
	Landmark Bias	1.1x map resolution
	Camera Pointing Knowledge Error	0.5° 3σ
	Image Matching Failure Rate	6.6%
	Camera Focal Length	0.006 m
	Minimum Map Resolution	1 m/pixel
	Pixels Per Axis	1024

MONTE CARLO ANALYSIS

A Monte Carlo analysis was conducted with the dispersions listed in Table 6 to assess the descent vehicle’s ability to land successfully given nominal conditions. Dispersions were used to create 2000 randomized cases which were ran using Simulink®’s rapid accelerator mode and parallel simulation capabilities. All dispersions assume Gaussian distributions.

Table 6. Vehicle Dispersions.

Category	Parameter	Dispersion

DESCENT ELEMENT		
Initial Conditions	Position	1927 / 1368 / 2331 m 3σ
	Velocity	1.71 / 2.09 / 1.22 m/s 3σ
	Body Rates	0.3°/s 3σ
	Attitude	3.0° 3σ
Propulsion	Peak thrust	Scale factor: 1% 3σ
	Peak ISP	Scale factor: 1% 3σ
Mass Properties	Mass	250 kg 3σ
	Center of gravity	0.05 / 0.01 / 0.01 m 3σ
	Moments of inertia	1% kg-m^2 3σ
	Products of inertia	1% kg-m^2 3σ
ASCENT ELEMENT		
Mass Properties	Mass	250 kg 3σ
	Center of gravity	0.05 / 0.01 / 0.01 m 3σ
	Moments of inertia	1% kg-m^2 3σ
	Products of inertia	1% kg-m^2 3σ

Table 7. Navigation and Sensor Dispersions.

SENSOR	***PARAMETER***	***VALUE***
IMU	Accelerometer Misalignment	17 arcsec 3σ/axis
	Accelerometer Scale Factor	450x0.1* ppm 3σ/axis
	Accelerometer Bias	84x0.1* micro-g 3σ/axis
	Accelerometer Velocity Random Walk	18 mm/s/$\sqrt{hr}$ 3σ/axis
	Gyroscope Misalignment	19 arcsec 3σ/axis
	Gyroscope Scale Factor	27x0.1* ppm 3σ/axis
	Gyroscope Bias	0.036x0.1* °/hr 3σ/axis
	Gyroscope Angular Random Walk	0.015 °/$\sqrt{hr}$ 3σ/axis
STAR TRACKER	Misalignment	8 arcsec 3σ
	Around/Cross Boresight Noise	24 arcsec 3σ
DSN	Position Error (u-v-w frame)	250 / 500 / 100 m 3σ

	Velocity Error (u-v-w frame)	0.025 / 0.05 / 0.005 m/s 3σ
NDL	NDL Error Model	NDL Model[18]
TRN	TRN Error Model	Specifications in Table 5
NOTE	* IMU bias and scale factors for both the accelerometer and gyro channels are scaled by 0.1 to represent an on-orbit calibration of the sensor prior to the DOI burn.	

The Monte Carlo simulation starts at 300 seconds before DOI and finishes at vehicle touchdown. Monte Carlo altitude plots from simulation start to touchdown for the 2000 runs are shown in Figure 9 and Figure 10, with zoomed-in altitude plots from PDI to touchdown shown in Figure 11 and Figure 12. The altitude time histories, Figure 9 and Figure 11, are overlaid with colored boxes that indicate when a given navigation sensor was active, approximately.

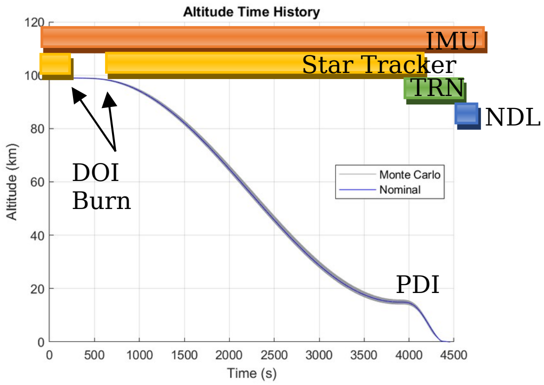


Figure 9. Altitude Time History.

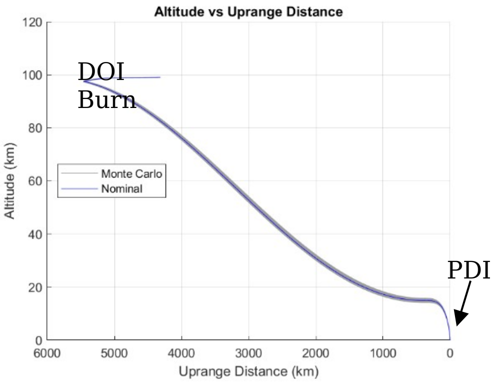


Figure 10. Altitude vs. Up range Distance.

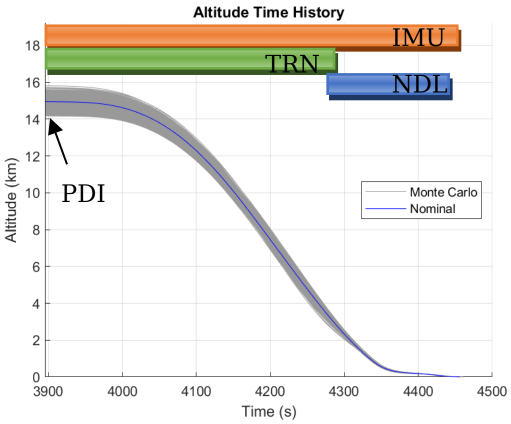


Figure 11. Altitude Time History (PDI to Touchdown)

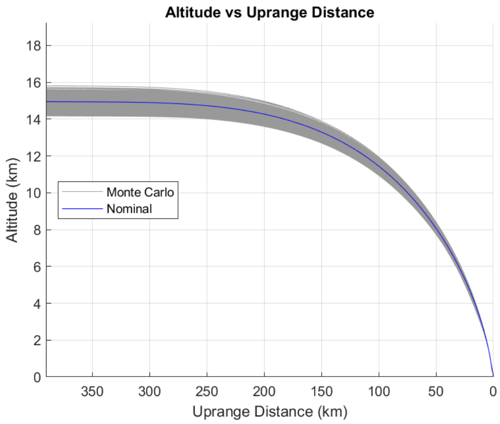


Figure 12. Altitude vs. Up range Distance (PDI to Touchdown)

As can be seen in Figures Figure 9 and Figure 10, the altitude dispersion widens during the post-DOI coast phase as navigation errors from the DOI burn propagate. However, prior to PDI, the TRN system comes online and gives the navigation system its first state observations since the DSN state update prior to DOI. Therefore, post PDI, the position dispersions converge (Figure 11 and Figure 12) as the vehicle performs its braking burn, approach burn, due to the contributions of the TRN system through the breaking burn and subsequently the NDL in the approach phase.

The landing success criteria used for this study is listed in Table 8. Using the criteria in Table 8, the 2000 Monte Carlo runs were assessed for successful landing. Specifically, requirements were compared against Monte Carlo statistics having a 99.73% probability of success with a 10% consumer risk. For a 2000 run Monte Carlo, this means interpolating each metric to its 1997.94th highest value, as derived from Hanson et al.[16]

Table 8. Landing Success Criteria.

TOUCHDOWN PERFORMANCE METRICS	CRITERIA
DISTANCE FROM TARGET	*< 100 m*
VERTICAL VELOCITY	*< 2 m/s*
HORIZONTAL VELOCITY	*< 1 m/s*
TILT ANGLE (RELATIVE TO LOCAL LEVEL)	*< 5°*
ATTITUDE ANGULAR RATE (RSS)	*< 0.5 °/s*

Table 9 shows the statistical results of the Monte Carlo for the successful landings. “Actual Landing” refers to true lander state at touchdown with respect to landing site target. Navigation errors are the difference between on-board navigation states (estimated position, velocity, attitude, and angular rate) and actual landing states (true position, velocity, attitude, and angular rate).

Table 9. Landing Success Statistics.

TOUCHDOWN PERFORMANCE METRIC	ACTUAL LANDING (99.73% PROB OF SUCC, 10% CR)	NAVIGATION ERRORS (99.73% PROB OF SUCC, 10% CR)
POSTION FROM TARGET	*38.7 m*	*38.1 m*
VERTICAL VELOCITY	*1.92 m/s*	*0.02 m/s*
HORIZONTAL VELOCITY	*0.62 m/s*	*0.02 m/s*
TILT ANGLE (RELATIVE TO LOCAL LANDING SITE FRAME)	*2.95 °*	*0.01 °*
ANGULAR RATE (RSS)	*0.37 °/s*	*<<0.01 °/s* *
***NOTE:** Typical navigation-grade IMU rate error is approximately +/- 0.02 deg/hr (~6x10^{-6} deg/sec) accounting for a calibration period prior to the DOI burn. This value is a composite of time constant and time varying error sources but disregarding the effects of quantization.		

Out of the 2000-case Monte Carlo, a single case was found to substantially exceed both the positional and velocity touchdown success criteria. However, because the 1997.94th highest value was taken for this Monte Carlo, this single case is indirectly discarded as an outlier and has no bearing on the results of Table 9. Since the single unsuccessful landing case is a statistical outlier, it is not shown in the dispersion plots that follow in this section.

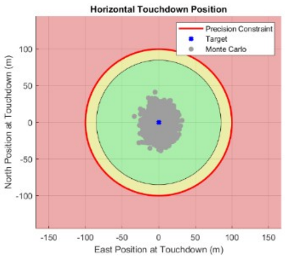


Figure 13. Actual Target Relative Position at Touchdown.

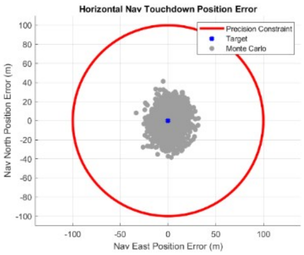


Figure 14. Navigation Position Error at Touchdown.

The achieved position dispersion at touchdown for the 1999 successful landings is shown in Error: Reference source not found, alongside the contributions due to navigation uncertainties in Error: Reference source not

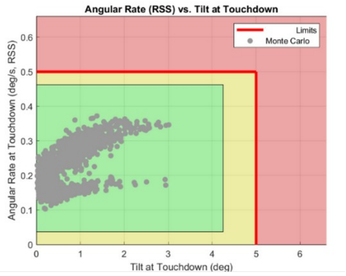


Figure 15. Actual Attitude at Touchdown.

Figure 17. Actual Velocity at Touchdown.

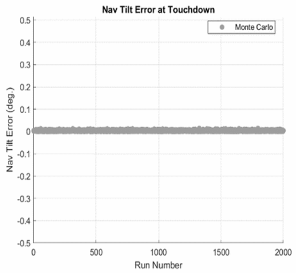


Figure 18. Navigation Attitude Error at Touchdown.

Figure 18. Navigation Velocity Error at Touchdown.

. Color coding in Error: Reference source not found, Error: Reference source not found, and Error: Reference source not found represent the success criteria in Table 8. Red means it violates the success criteria. Yellow means is within the success criteria of Table 8 but is within 15% of the success criteria margin, so “marginally successful.” And green means that the success criteria is met with a 15% of margin. Comparing Error: Reference source not found and Error: Reference source not found, as well as Table 9, it can be seen that the actual vehicle position error is completely driven by the navigation error. However, comparing actual touchdown velocity with its corresponding velocity navigation error, Error: Reference source not found and Error: Reference source not found, respectively, it can be seen that the velocity navigation errors are very small compared to the actual landing values. Therefore, velocities at landing are mostly driven by the Guidance and Control performance. Similarly, touchdown tilt angle versus angular rate is shown in Error: Reference source not found with the corresponding navigation error contributions shown in Error: Reference source not found (rate errors are negligibly small ~$6x10^{-6}$ deg/sec). Here, the attitude and attitude rate errors are very small compared with the actual attitude and attitude rates at touchdown. Therefore, the attitude and attitude rates at touchdown are mostly driven by the Guidance and Control performance.

Error: Reference source not found, Error: Reference source not found, and Error: Reference source not found show that the landing met the success criteria of Table 8. This type of landing performance assessment is repeated regularly as the design matures or different assumptions are applied. One such assumption is discussed in the next section, where concerns about the effect of lunar surface lighting on the TRN system’s performance prompt a study into just how late TRN can possibly get its first measurement fix and the vehicle still land successfully.

EXAMPLE NAVIGATION TRADE

As previously discussed, Terrain Relative Navigation (TRN) is a critical technology for enabling precision landing in the absence of a GPS-like system on the Moon. However, passive optical (camera based) TRN relies on lunar surface illumination for capturing useful images. But for certain landing locations and approach trajectories there could be low or no lighting in the camera field of view for all or a portion of the descent trajectory. Low or no illumination is especially prevalent at polar landing regions with shallow sun angles. NASA has recently identified and published 13 landing regions at the south pole.[*] Naturally, TRN images that have no illumination will not yield discernable surface features, and a passive optical TRN system will not be able to determine a position solution. Figure 19 shows an example of a polar landing descent trajectory, blue line represents trajectory

[*] “NASA Identifies Candidate Regions for Landing Next Americans on Moon” URL: https://www.nasa.gov/press-release/nasa-identifies-candidate-regions-for-landing-next-americans-on-moon

with direction from left to right, for different landing times at the Connecting Ridge landing site region. Depending on landing time, a descent trajectory to a polar landing site may only have illuminated terrain very late, near the landing site, as shown in Figure 19 (b) and (c).

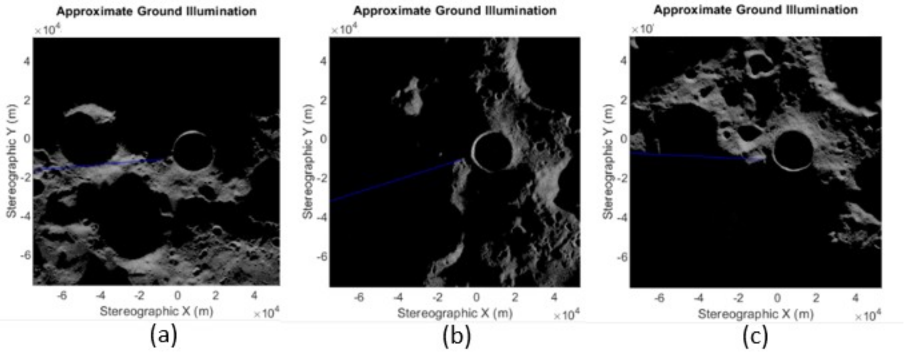

Approximate Ground Illumination
Approximate Ground Illumination
Approximate Ground Illumination
Stereographic Y (m)
Stereographic X (m)
×10⁴
(a)
(b)
(c)

Figure 19. Example Surface Illumination in a Polar Landing Site.

The purpose of this section is to discuss an example trade study that looks at how late in the descent trajectory TRN can obtain valid measurements and the vehicle still has enough time to make the necessary trajectory corrections and achieve a successful soft and precise landing.

With GLASS, setting up this type of a study is easy. One additional Monte Carlo dispersion is added to the baseline set. The TRN-on condition was varied to represent different moments when lighted terrain came into the camera view by dispersing maximum up-range distance. Maximum TRN up-range distance was dispersed uniformly between 5km to 250km. At the low end, TRN measurements coming late at only 5km up range from landing, the vehicle’s GNC has very little time to make the large amounts of horizontal position error corrections, causing high rates of unsuccessful landings. At the high end of 250km, TRN comes on prior to PDI, improving position navigation early and giving the vehicle plenty of time to fly a near-optimal descent trajectory with very accurate navigation knowledge. However, mission designers would be constrained to finding trajectories that guarantee lighting at 250 km up-range of the landing site.

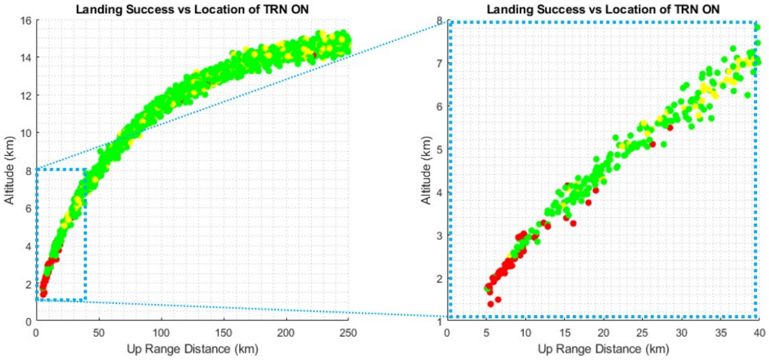


Figure 20. Landing Success vs. TRN First Measurements.

Figure 20, depicts landing success for a 2000 run Monte Carlo with the new TRN up range dispersion in addition to all those in Table 6 and Table 7. The horizontal and vertical axes represent the up range distance and altitude, respectively, at which TRN was enabled. The color coding on each marker indicates three possible types of landing: a landing comfortably within requirements (green), a landing marginally within requirements (yellow), or an unsuccessful landing (red). An unsuccessful landing is one which violates at least one success criteria in Table 8. A marginally successful landing is one which is within the success criteria of Table 8 but comes within 15% of one or more of the success criteria. A comfortably successful landing is one below 15% for all success criteria. On Figure 20, it should be noted that TRN availability is not the only driver in landing success, and there are very few outlier failure cases driven by the effect of other dispersions despite favorable TRN availability. Looking at the data in another way, Figure 21, shows percent landing success as a function of TRN first measurement up range distance.

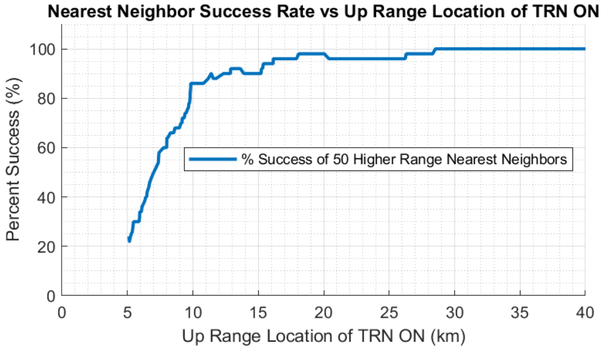


Figure 21. Landing Success vs. First TRN Measurements Up range Distance.

Therefore, **Figure 21,** shows how far up range the first TRN fix must be to have a high likelihood of success. A successful landing becomes very likely for this vehicle if the TRN system acquires a valid measurement at least 30 km up range from the landing site. To confirm the 30 km up range minimum illumination threshold, future detail work outside the scope of this paper would focus on detail Monte Carlo analysis with dedicated TRN on-time fixed at 30km up range. This type of analysis could be used to drive mission design illumination constraints, i.e. including how far up range a candidate trajectory must guarantee terrain lighting, and at which altitude the TRN system must be capable of acquiring the first measurements. In addition, Figure 21 shows rapid landing degradation performance, with success rates less than 80%, for TRN acquisition at or less than a10 km up range from target, approximately. This is just one example of how simulations using GLASS can be used to inform the impact of mission constrains, such as surface illumination on landing success for this baseline set of GNC sensor suite.

CONCLUSIONS

The NASA Marshall Space Flight Center has developed the GeneraLized Aerospace Simulation in Simulink® (GLASS) tool. This tool allows the rapid development and evaluation of guidance, navigation, and control performance for a single or multi-element vehicle architecture. GLASS is built in the MathWorks® Simulink® environment language which provides a model-based design framework, highly suitable for integration of vehicle models in a modular fashion. The Simulink® environment allows seamless use of all the MathWorks® capabilities and toolboxes, including quick generation of flight code though auto-coding, and software and hardware in the loop testing capabilities. Using GLASS, the GNC performance for a government reference design of the Human Landing System has been evaluated for nominal and dispersed Monte Carlo cases. In addition, the GLASS tool was used to perform an example navigation trade that showed how the vehicle landing performance varied depending on when the Terrain Relative Navigation sensor became operational during the descent phase, representing potential lunar surface illumination constraints. This type of trade allows NASA to understand the robustness of a given GNC architecture subject to potential mission constraints.

REFERENCES

1. J. Orphee, M. Hannan, N. Ahmad, E. Anzalone, E. Braden, S. Craig, J. Everett, K. Miller, N. Olson, “Guidance, Navigation, and Control for NASA Lunar Pallet Lander” 42nd Annual American Astronautical Society Guidance and Control Conference; Feb 1 – Feb 6 2019; Breckenridge, Colorado, AAS 19-033.

2. E. Anzalone, E. Braden, N. Ahmad, D., K. Miller, “Guidance and Navigation Design Trades for the Lunar Pallet Lander” 42nd Annual American Astronautical Society Guidance and Control Conference; Feb 1 – Feb 6 2019; Breckenridge, Colorado, AAS 19-091.

3. S. Craig, J. Holt, M. Hannan, J. Orphee “Mission Design for the Lunar Pallet Lander” 42nd Annual American Astronautical Society Guidance and Control Conference; Feb 1 – Feb 6 2019; Breckenridge, Colorado, AAS 19-330.

4. Craig, A. S., Anzalone, E. J., Hannan, M. R. , Belanger, B. L., Burke, L. M., Condon, G. L., Joyce, R. T., Mahajan, B., Summers, A. W., Means, L. L., and Pei, J., “Human Landing System Storable Propellant Architecture: Mission Design, Guidance, Navigation, and Control,” AAS 20-592.

5. Acton, C.H.; "Ancillary Data Services of NASA's Navigation and Ancillary Information Facility;" Planetary and Space Science, Vol. 44, No. 1, pp. 65-70, 1996.

6. Wood, G., Kennedy, D., (2003) “Simulating Mechanical Systems in Simulink with SimMechanics.” Technical Report of The Mathworks Inc., Available from www.mathworks.com.

7. Walton Jr., W.C., Herr, R.W., and Leonard, H.W., "Studies of Touchdown Stability for Lunar Landing Vehicles," J. spacecraft, Vol. 1, No. 5,1964.

8. Roncoli, R., “Lunar Constants and Models Document.” Jet Propulsion Laboratory, September 23, 2005.

9. Lu, P., “Theory of Fractional-Polynomial Powered Descent Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 3, March 2020, pp. 398-409

10. Orr, Slegers. “High-Efficiency Thrust Vector Control Allocation”. JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS. DOI: 10.2514/1.61644,

11. Hall, Hough, Orphee, Clements. "Design and Stability of an On-Orbit Attitude Control System Using Reaction Control Thrusters”.

12. Shidner, J., “An Efficient Ray-Tracing Method for Determining Terrain Intercepts in EDL Simulations,” 2016 IEEE Aerospace Conference, 2016, pp. 1-9, doi: 10.1109/AERO.2016.7500591.

13. Gragossian, A., Pierrottet, D., Estes, J., Barnes, B. W., Amzajerdian, F., and Hines, G. D., “Navigation Doppler Lidar Performance at High Speed and Long Range,” AIAA 2020-0369.

14. Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L., “Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing,” IEEE Transactions on Robotics, vol. 25, no. 2, pp. 264-280, 2009.

15. Johnson, A., Aaron, S., Chang, J., Cheng, Y., Montgomery, J., Mohan, M., Schroeder, S., Tweddle, B., Trawny, N., and Zheng, J., “The Lander Vision System for Mars 2020 Entry Descent and Landing,” Proceedings of the Advances in the Astronautical Science Guidance Navigation and Control, vol. 159, no. 36, 2017

16. Hanson, J. M., Beard, B. B., “Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Analysis,” AIAA Journal of Spacecraft and Rockets, vol/ 49, no. 1, 2012